VIDEO BASIC

20 VIDEOLEZIONI DI BASIC
PER IMPARARE COL VIC 20

- Le periferiche del VIC 20
- Studiare il problema
- I diagrammi a blocchi
- Scrivere il programma
- INPUT, LIST, RUN, END
- Il tuo primo programma
- Gli operatori aritmetici
- Videogioco n. 2

GRUPPO
EDITORIALE
JACKSON

2

COMMODORE
VIC 20

VIDEO BASIC
Pubblicazione a fascicoli quattordicinali
edita dal Gruppo Editoriale Jackson
Direttore Responsabile:
Giampietro Zanga
Direttore e Coordinatore
Editoriale: Roberto Pancaldi
Autore: Softidea
Redazione software:
Francesco Franceschini, Enrico Braglia,
Fabio Calanca
Segretaria di Redazione:
Marta Menegardo
Progetto grafico:
Studio Nuovaidea - Via Longhi 16 - Milano
Impaginazione:
Silvana Corbelli
Illustrazioni:
Cinzia Ferrari, Silvano Scolari
Fotografie:
Marcello Longhini
Distribuzione: SODIP
Via Zuretti, 12 - Milano
Fotocomposizione: Lineacomp S.r.l.
Via Rosellini, 12 - Milano
Stampa: Grafika '78
Via Trieste, 20 - Pioltello (MI)
Direzione e Redazione:
Via Rosellini, 12 - 20124 Milano
Tel. 02/6880951/5

Autorizzazione alla pubblicazione Tribunale di
Milano n° 422 del 22-9-1984
Spedizione in abbonamento postale Gruppo II/70
(autorizzazione della Direzione Provinciale delle
PPTT di Milano).
Prezzo del fascicolo L. 8.000
Abbonamento comprensivo di 5 raccoglitori L. 165.000
I versamenti vanno indirizzati a: Gruppo
Editoriale Jackson S.r.l. - Via Rosellini, 12
20124 Milano, mediante emissione di assegno
bancario o cartolina vaglia oppure
utilizzando il c.c.p. n° 11666203.
I numeri arretrati saranno disponibili per un anno
dal completamento dell'opera e potranno essere
prenotati presso le edicole o richiesti direttamente
alla casa editrice. Ai fascicoli arretrati verrà
applicato un sovrapprezzo di L. 400 sul prezzo di
copertina (a partire dalla 12ª settimana di distribuzione).
Non vengono effettuate spedizioni contrassegno.

Gruppo Editoriale Jackson

Direzione Editoriale
Giampietro Zanga
e Paolo Reina

SOMMARIO

Introduzione

*Dopo aver esaminato nella prima lezione di VIDEOBASIC lo schema di un computer, nelle prossime pagine tratteremo delle varie unità complementari che possono essere collegate al tuo VIC 20.
Nella parte dedicata alla teoria scopriremo come il BASIC utilizzi alcuni simboli delle operazioni aritmetiche, diversi da quelli che usiamo comunemente.
Poi, dopo aver fatto conoscenza delle nuove istruzioni e dei nuovi comandi, saremo finalmente in grado di progettare e di scrivere il nostro primo programma.*

Le periferiche del VIC 20

Come già detto la parte fondamentale e caratterizzante del VIC 20 è l'unità centrale, il cui compito, analogo a quello della mente umana, è quello di elaborare i dati.
Sono però necessari dei dispositivi in grado di fornire i dati alla "mente" del computer ed altri che rendano disponibili ed usufruibili i risultati dell'elaborazione.
Il nostro cervello, pur in grado di fare mille cose,

1) Paddle
2) Joystick
3) Tavoletta grafica
4) Penna ottica
5) Registratore
6) Floppy disk drive
7) Modem
8) Televisore
9) Monitor
10) Stampante
11) Plotter
12) Robot

Dall'1 al 4 sono dispositivi di INPUT.
Dall'5 al 7 sono dispositivi sia di INPUT che di OUTPUT.
Dall'8 al 12 sono solo di OUTPUT.

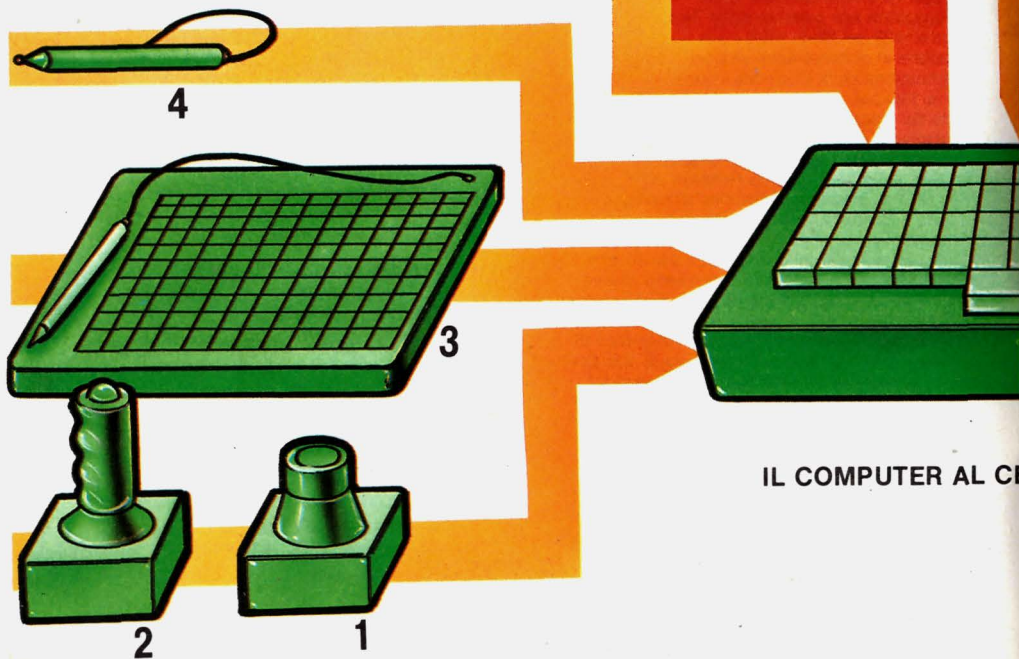
5

6

4

3

2

1

HARDWARE

ha bisogno del corpo per mettere in atto le sue capacità.
Allo stesso modo, per poter operare in concreto, l'unità centrale necessita di "braccia" o di "occhi": le periferiche.
Questo nome deriva dal fatto che solitamente circondano l'unità centrale alla quale sono quasi sempre fisicamente collegate.
Possono essere di INPUT quando trasmettono dati alla CPU, di OUTPUT quando li ricevono ed anche sia di INPUT che di OUTPUT, quando entrambe le funzioni vengono espletate dallo stesso dispositivo.

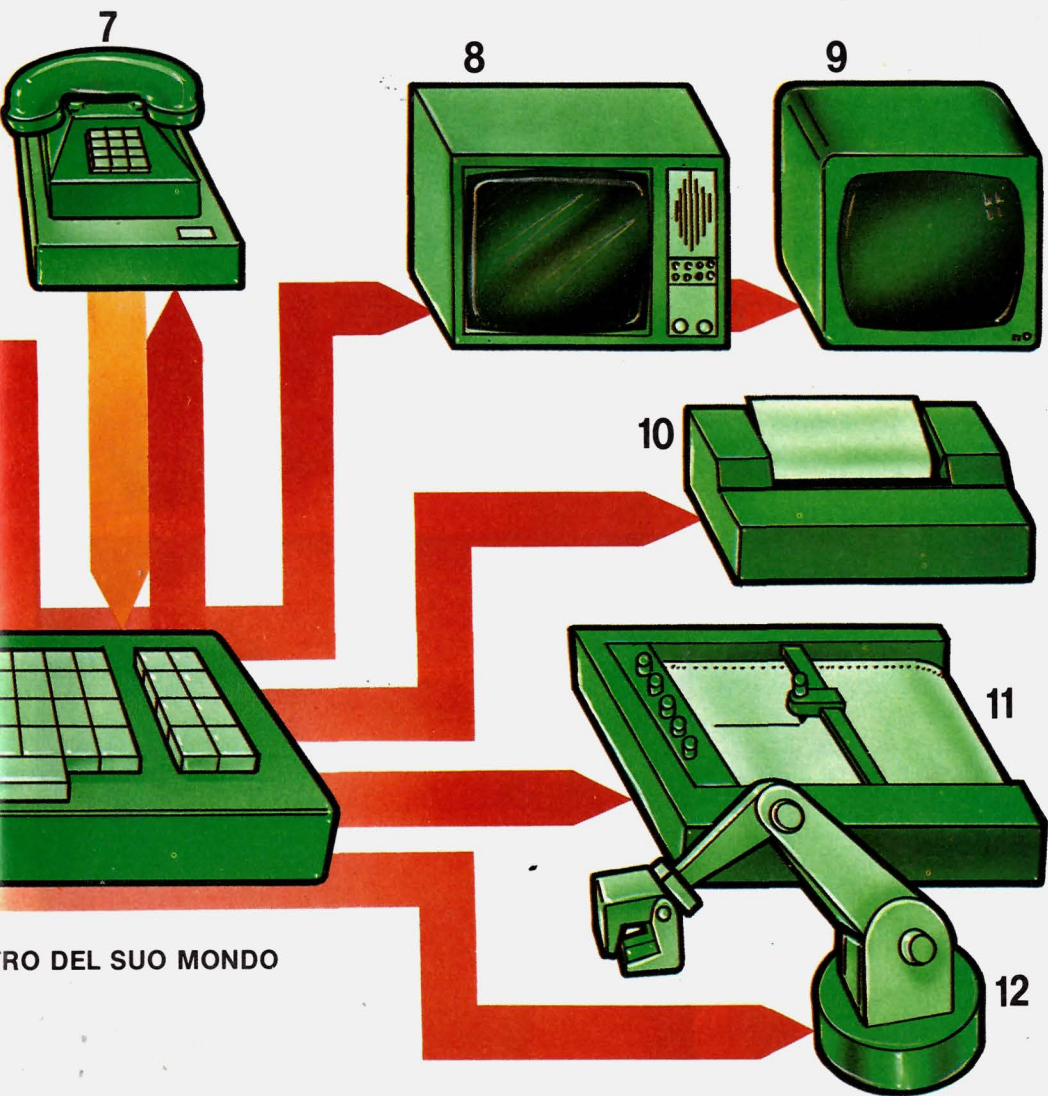
7

8

9

10

11

12

RO DEL SUO MONDO

HARDWARE

TV e monitor

Sia la TV che il monitor sono apparecchiature destinate a visualizzare il segnale video proveniente dal tuo VIC 20.
La prima è un normale televisore domestico in cui al posto dell'antenna colleghi, tramite l'apposito cavetto, l'uscita del segnale video del computer.
Il segnale viene adeguatamente decodificato (trasformato) dalla TV che riproduce sullo schermo l'immagine contenuta nella memoria grazie alla scansione effettuata da un fascio di elettroni.
Quando gli elettroni colpiscono i fosfori (deposti sulla superficie interna dello schermo), questi emettono luce riproducendo per punti l'immagine.
Il monitor è la naturale evoluzione del televisore e, rispetto a quest'ultimo, si differenzia per il processo di decodifica delle immagini, che ne consente una migliore qualità.
Grazie alla maggiore stabilità e nitidezza, infatti permette una lettura più riposante delle informazioni ed è per questo consigliabile per chi opera professionalmente davanti ad un terminale video. Per un uso hobbistico del computer, invece, è più che sufficiente un buon televisore.

HARDWARE

Registratore

Viene usato come memoria di massa, ovvero come un contenitore dove vengono immagazzinate, senza perderle, una grande quantità di informazioni.
Queste vengono *convertite dal computer* in suoni secondo uno schema definito ed in questa forma incise su nastro magnetico alla stregua di un qualsiasi segnale audio.
Il registratore è uno degli elementi più delicati del sistema: se qualche sua parte non funziona, non riesci a caricare le informazioni precedentemente memorizzate su nastro.
Per questo motivo è consigliabile prestare la *massima cura nell'uso di* questa unità.
In particolare, è buona norma pulire periodicamente la testina di registrazione. Eviterai così distorsioni che ingannano il computer inviandogli informazioni sbagliate.
È pure consigliabile registrare a un volume elevato in modo che il segnale sia facilmente "udibile" dal computer.
È raccomandabile, infine, utilizzare il contagiri e prender nota dei nomi e della posizione dei programmi presenti nella cassetta, in modo da poterli rintracciare rapidamente.

Floppy disk drive

Questa unità svolge lo stesso compito del registratore e perciò i floppy disk (o dischi flessibili) sono anch'essi delle memorie di massa.
La differenza principale, che caratterizza i floppy disk dai registratori, è la velocità con cui vengono caricate e memorizzate le informazioni.
Infatti la velocità di caricamento varia da un minimo di 2 a un massimo di 8/10 volte quella del registratore.
Le dimensioni dei dischetti sono, normalmente, circa quelle di un disco a 45 giri (esattamente 5 pollici 1/4).
Vi sono, però, anche diversi formati a seconda del drive usato dal computer: 3-3, 5-8 pollici.
Un'altra importante differenza col registratore consiste nel procedimento di ricerca di una informazione.
Nei dischetti, infatti, è possibile la ricerca sia in avanti che indietro poiché la testina può muoversi radialmente, al contrario di quanto accade con il registratore dove è il nastro che deve scorrere davanti alla testina che è fissa.
Si dice che il disco è una memoria di massa a ricerca casuale, poiché permette la ricerca del dato direttamente nella posizione in cui è stato memorizzato.
Il nastro invece, per il fatto che può andare solo in avanti mentre carica dei dati, si dice che è una memoria di massa sequenziale.
Infatti sul nastro le informazioni sono memorizzate una di seguito all'altra cioè in sequenza.

Stampante

La stampante è un dispositivo in grado di imprimere su carta le informazioni elaborate (numeri e testi).

Nella stragrande maggioranza dei casi il suo funzionamento è sostanzialmente simile a quello di una macchina per scrivere: una testina batte il carattere voluto su di un nastro inchiostrato a contatto del foglio.
Le stampanti più diffuse adottano delle testine ad aghi i quali, spinti da martelletti, formano per punti il carattere da stampare.
Nelle stampanti a margherita, invece, i caratteri sono già presenti ognuno sulla propria asta esattamente come in una comune macchina per scrivere.
Le stampanti ad aghi sono in genere veloci (possono imprimere da 30 a oltre 400 caratteri in un secondo) e relativamente economiche.
Quelle a margherita sono più lente (da 10 a 50 caratteri per secondo) e più costose, ma consentono una migliore qualità di stampa.

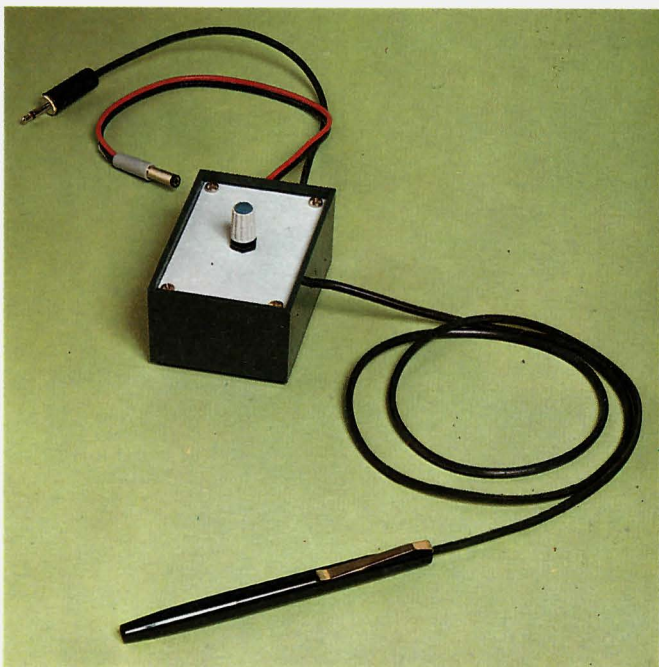
Plotter

Le stampanti hanno alcune limitazioni nell'uso grafico: precisione non troppo elevata (l'immagine è costituita da singoli punti) e dimensioni limitate del foglio.
Quando occorre, perciò, stampare su carta o lucido un disegno si ricorre al plotter. Si tratta sostanzialmente di un tecnigrafo automatico costituito da un pennino scorrevole lungo due guide libere di spostarsi.
A seconda se il pennino è alzato e abbassato esso traccerà un punto se le guide sono ferme, una linea se sono in movimento. In questo *semplice modo* il plotter è in grado di realizzare disegni molto complessi.
Non solo.
Il plotter può scrivere testi come una qualsiasi stampante, senza però il limite costituito dal tipo di testina: è in grado di stampare qualsiasi carattere in qualsiasi stile.

Tavola grafica

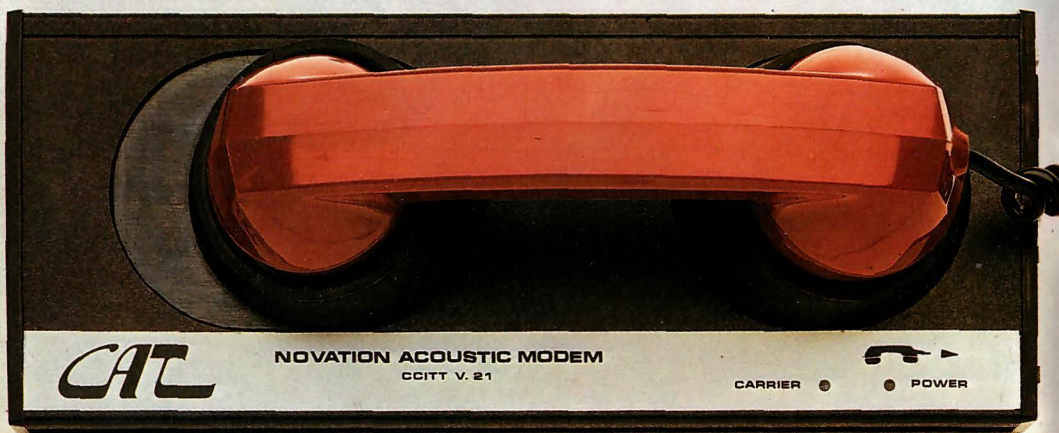
È una sorta di lavagna elettronica dove il gesso è sostituito da una penna speciale e la lavagna da un piano sensibile all'azione della penna. È in altre parole il reciproco del plotter: si tratta sempre di un "tecnigrafo" destinato però a comunicare informazioni al computer e non a visualizzare quelle ricevute dalla macchina. Il suo uso è lo stesso di una lavagna tradizionale ma con numerosi vantaggi rispetto ad essa: possibilità di cancellare facilmente, in tutto o in parte, quello che si è fatto; possibilità di poter costruire figure geometriche con l'aiuto del computer semplicemente posizionando la penna in un modo particolare; possibilità di correggere molto semplicemente ogni disegno; possibilità di colorare una determinata area.
La tavoletta grafica (o digitizer) è uno strumento preziosissimo per chiunque si interessi professionalmente di grafica (ingegneri, architetti, grafici, stilisti...); per la sua semplicità e potenza rende sorpassati i precedenti tavoli da disegno.

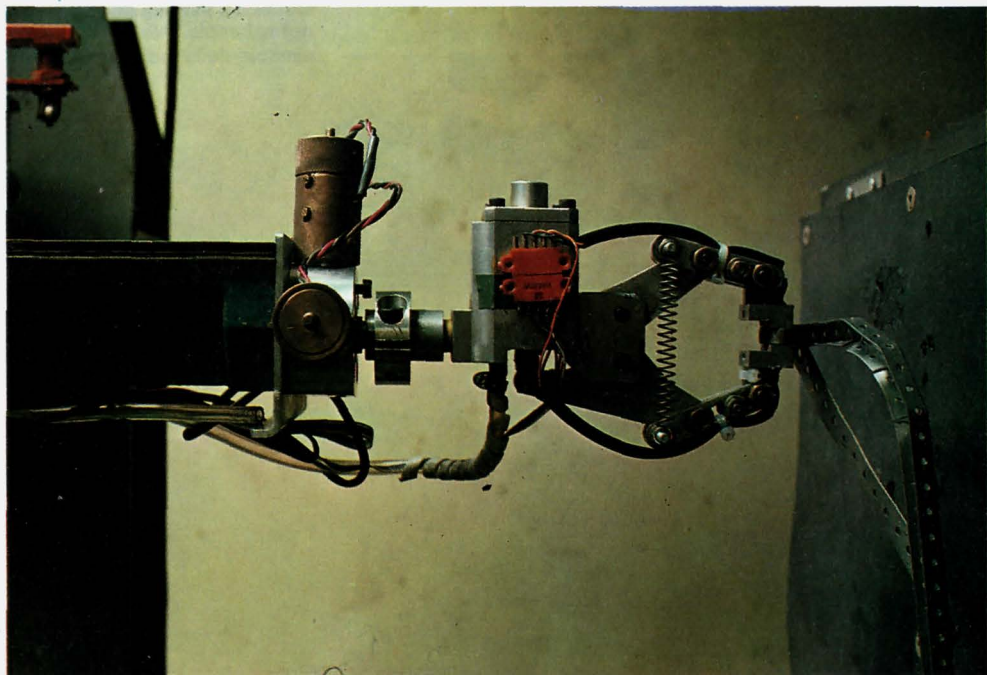
Penna ottica

La penna ottica è l'analogo, per il video, della penna della tavoletta grafica: assieme al video ha, più o meno, le stesse funzioni del digitizer con (rispetto a questo) una minore precisione, ma una maggiore economicità.
Il principio di funzionamento della penna ottica, o light pen, è piuttosto semplice.
Ricorderai che l'immagine di un televisore o monitor è costituita da punti luminosi dovuti alla eccitazione dei fosfori.
Quando la penna, alla cui estremità è montato un sensore ottico, rileva della luminosità, vuol dire che il fascio di elettroni sta colpendo i fosfori immediatamente sottostanti. A questo punto, appositi circuiti determinano le coordinate esatte del punto dove è appoggiata la penna.

Joystick

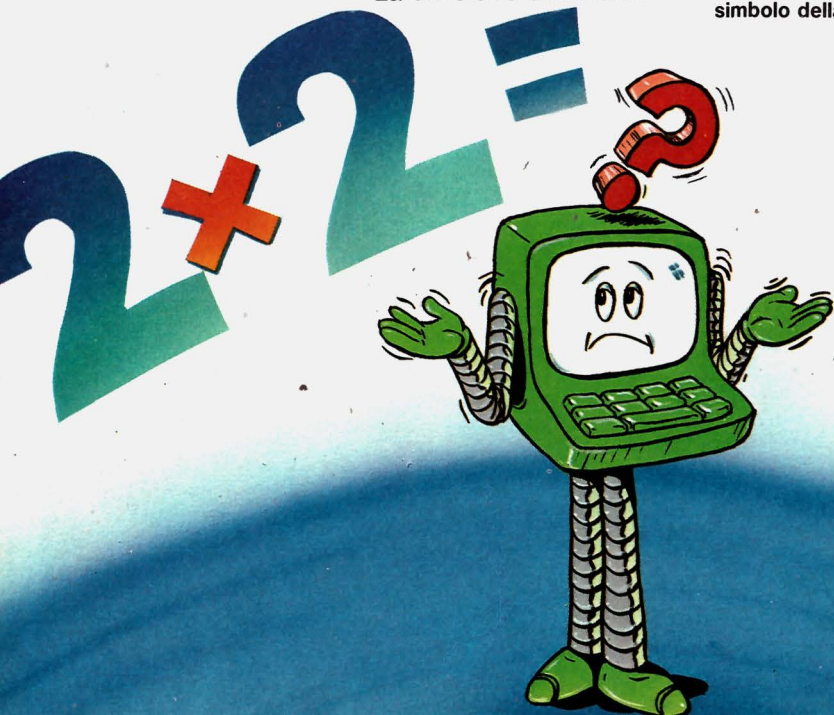
Il joystick (una sorta di elementare tastiera esterna) è un dispositivo di input costituito essenzialmente da una leva orientabile in almeno 4 direzioni (alto, basso, destra e sinistra) con la quale puoi dare semplici informazioni di posizione al computer. Conosciuto e usato da tutti gli appassionati di videogiochi, consente di spostare in modo immediato un determinato elemento sullo schermo. Vantaggio non indifferente rispetto alla tastiera tradizionale è la possibilità di unire, ad un comando come per esempio alza il cursore, l'azione fisica di spostare con la mano la leva verso l'alto, rendendo facile e naturale, oltre che comodo, l'uso di questa unità. L'unica avvertenza è che non tutti i joystick esistenti sono collegabili agli home computer, quindi è consigliabile informarsi su quali possono essere utilizzati sul tuo VIC 20. Per la sua comodità è un accessorio che aumenta notevolmente l'economia del sistema (cioè la facilità d'uso da parte di una persona).

Modem

È l'acronimo di modulatore/demodulatore, cioè una unità in grado di convertire i dati e i programmi in forma tale che possano essere trasmessi via cavo telefonico e, naturalmente, di compiere l'operazione opposta cioè "tradurre" i segnali provenienti dalla linea telefonica in dati utilizzabili. Consente quindi di collegare un calcolatore ad un altro, mettendo in comune dati e programmi anche a distanze notevoli in brevissimo tempo.
Su questo principio si fondano le banche dati, cioè enormi archivi elettronici contenenti informazioni di vario genere messi a disposizione degli utenti. Ad esempio puoi sapere tutti i libri su un determinato argomento pubblicati in Italia, o prenotare una stanza, stando comodamente a casa.
Un tipo di modem a basso costo è l'accoppiatore acustico utilizzabile con tutti gli home e personal computer.
Una volta collegato al tuo VIC 20 il suo uso è molto semplice: basta appoggiare il ricevitore telefonico nell'alloggiamento dell'unità, abilitare il sistema a trasmettere o ricevere dati, formare il numero telefonico corrispondente al computer da chiamare, e infine trasmettere (o ricevere) i dati.

Robot

Il nome Robot (derivato dal *cecoslovacco* Robota) venne coniato dal commediografo A. Capevi per definire, nell'ambito di una sua commedia, dei servitori *meccanici*. La fantascienza si è poi impadronita del termine per indicare le macchine dall'aspetto e comportamento umani. In realtà un robot è una *unità elettronica-meccanica* (ben diversa dall'uomo), pilotata da un computer, in grado di compiere determinate azioni (trasportare materiali, montare pezzi *meccanici*, verniciare...) in sostituzione dell'uomo nelle attività pericolose, ripetitive o di estrema *precisione*.
Nel caso di un robot per home computer, come il tuo VIC 20, il suo aspetto è il più delle volte quello di un braccio meccanico capace di compiere (per ora) *semplici operazioni* grazie ad opportuni programmi inseriti nel computer.
Per concludere. Il robot è la propagazione "fisica" del computer verso il mondo esterno che gli permette di tramutare in azione i dati elaborati.

Operatori aritmetici

Le espressioni matematiche, in BASIC, sono praticamente identiche a quelle che hai studiato a scuola, con alcune piccole differenze nella scrittura degli operatori algebrici (i simboli delle operazioni).
In BASIC le operazioni di somma e di sottrazione si indicano come al solito con "+" e "—".
Il prodotto viene invece indicato con l'asterisco "*" e non con "x" o il punto.
Spesso in matematica il prodotto di due valori si indica semplicemente scrivendoli uno dopo l'altro; così "a b" vuol dire "a per b". Per il tuo VIC 20, invece, nulla è mai sottinteso, per cui il prodotto deve essere espressamente indicato con l'asterisco: A*B (nota che in BASIC tutto va in lettere maiuscole).
La divisione è indicata con il carattere "/" (barra, in inglese slash); i due punti (:) infatti verrebbero interpretati dal tuo VIC 20 come interpunzione.
L'elevamento a potenza è un'altra operazione con un simbolo particolare. Non si scrive l'esponente come un numero più piccolo posto in alto a destra della base come "4^3". Si scrivono invece base ed esponente sulla stessa riga, separato da uno "↑", per il nostro esempio 4↑3.

Il tuo VIC 20 non riconosce alcuni dei simboli che usi normalmente. Per lui * è il simbolo della moltiplicazione.

Gli operatori relazionali

Gli operatori di relazione sono simboli che servono per stabilire un ordine di grandezza fra due elementi. I più comuni sono $=$; $\neq$; $>$; $<$.

a) Il simbolo $=$ sta a rappresentare che due o più grandezze sono legate da una relazione di identità. Per esempio,

□ = □;

$4 = 4$;

$4 = 4$
4 uguale a 4

b) Il suo contrario è la relazione di disuguaglianza, rappresentata dal simbolo matematico $\neq$, relazione che è molto più facile da individuare, sul piano strettamente numerico, rispetto alla prima. Difatti sono infinitamente più numerose le grandezze disuguali fra loro, di quelle uguali

□ ≠ Δ

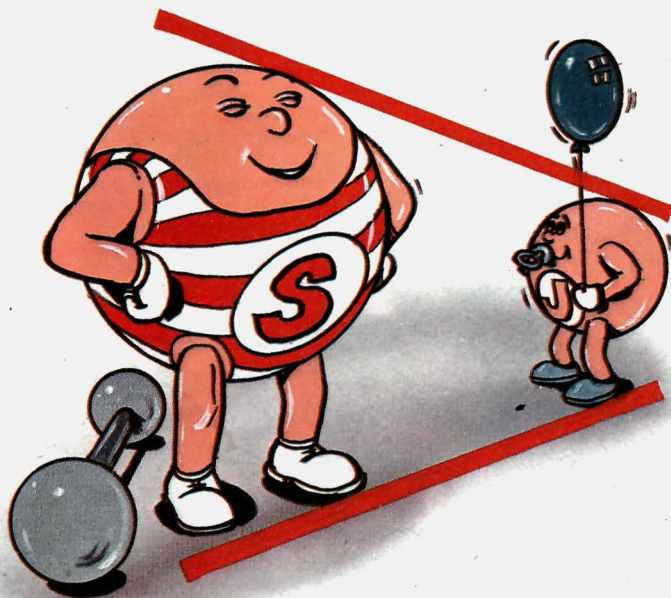
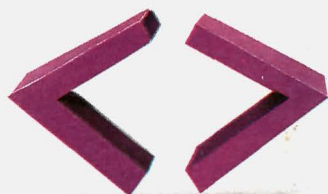
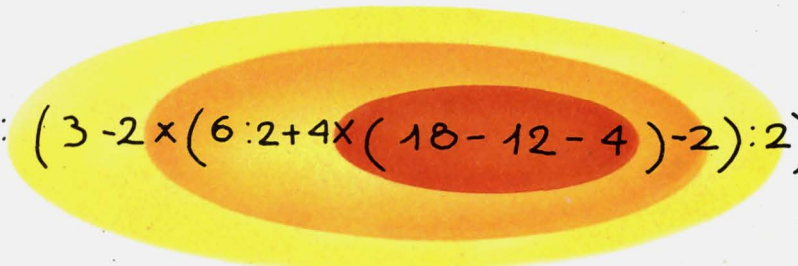
$2 \neq 3$

2 è diverso da 3

c) Due grandezze diverse possono a loro volta essere maggiori o minori l'una rispetto l'altra. Questi altri due tipi di relazione sono espressi da simboli maggiore ($>$) e minore ($<$).

$6 > 2$
6 maggiore di 2

$2 < 8$
2 minore di 8

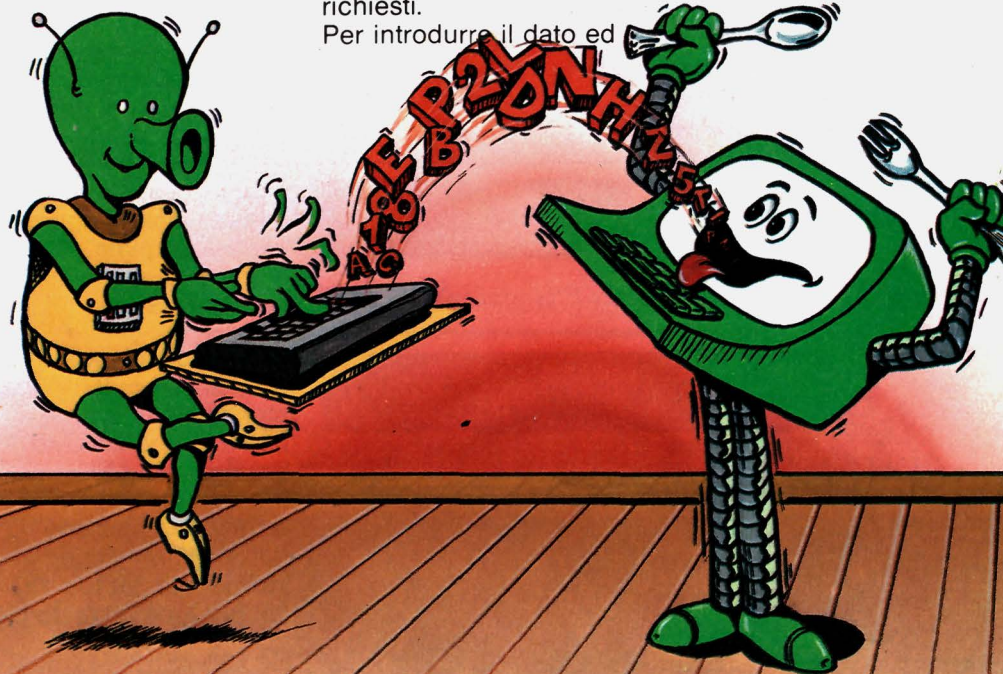
N.B. Per i computer il concetto di disuguaglianza è rappresentato, sul piano della soluzione grafica, dal simbolo $< >$.

Le parentesi e la loro priorità

In matematica esistono dei simboli, detti parentesi, usati nel calcolo di espressioni, aritmetiche o algebriche, che possono racchiudere alcuni elementi delle espressioni stesse.
Le operazioni tra parentesi vanno eseguite prima delle altre.
Esistono tre tipi di parentesi che determinano l'ordine di esecuzione delle operazioni e sono: la parentesi tonda ( ), la parentesi quadra [ ] e quella graffa { }.
Esse indicano in ordine di successione la precedenza di una singola operazione.
Il BASIC, invece, conosce solo la parentesi tonda. Ne puoi mettere quante vuoi, *l'imporante è chiuderle!*
Il tuo VIC 20 inizierà a calcolare partendo dalla parentesi tonda più *interna per proseguire* via via verso l'esterno.
Riassumiamo ora in ordine di priorità gli operatori algebrici e relazionali:
1) elevamento a potenza;
2) moltiplicazione - divisione;
3) somma-sottrazione;
4) uguale - maggiore - minore.

Il tuo VIC 20 inizia a calcolare dalla parentesi più interna, risolvendo prima l'espressione (18 − 12 − 4) proseguendo poi con quelle più esterne.

$$12 : (3 - 2 \times (6 : 2 + 4 \times (18 - 12 - 4) - 2) : 2) + 14$$

INPUT

Ci sono molto spesso casi in cui chi scrive il programma non conosce i dati che dovranno essere elaborati.
Supponiamo, ad esempio, che un tuo prossimo programma abbia come obiettivo quello di calcolare l'età media delle prime 5 persone che lo utilizzeranno.
Dovrà quindi operare su dati (l'età di quelle persone) che non ti sono noti.
L'unico ragionevole modo di venire in possesso dei dati necessari è quello di chiederli direttamente a chi è in grado di fornirli, cioè a quelle persone.
Questo è il compito dell'istruzione INPUT.
Tramite essa puoi richiedere all'utente del programma l'introduzione da tastiera di uno o più dati; questi vengono memorizzati nella o nelle variabili che compaiono nell'istruzione.
Quando il programma incontra l'istruzione INPUT, il VIC 20 stampa un ? (punto di domanda) e si arresta fino a quando non sono stati forniti tutti i valori richiesti.
Per introdurre il dato ed assegnarlo alla relativa variabile, è necessario battere alla fine della sequenza di caratteri il tasto RETURN.
Se i dati da fornire sono più di uno, cioè se vi sono più variabili nell'istruzione INPUT, essi possono essere introdotti uno di seguito all'altro purché siano separati da una virgola. L'ultimo dovrà essere seguito da RETURN.
Oppure. Puoi fornire i dati singolarmente, ciascuno seguito da un RETURN, nel qual caso apparirà un doppio punto di domanda (??) ad indicarti che l'introduzione dei dati

non è completa.
Devi fornire un dato per ogni variabile contenuta nell'istruzione e il dato, cosa molto importante, deve essere dello stesso tipo della variabile: numero o stringa. Se il dato è di tipo testo e la variabile è numerica ti apparirà il messaggio:

```
?redo from start
```

Poiché il programma può richiedere molti dati in ingresso è indispensabile che il tuo VIC 20 segnali a te (o a chiunque stia usando il programma) quali dati devi inserire.
Ecco la necessità del messaggio da anteporre (tra virgolette) alla o alle variabili, la prima delle quali preceduta da ; (punto e virgola), contenute nell'istruzione INPUT.

```
INPUT A
```

"A, cosa rappresenta? La data di nascita? L'età? Il codice fiscale?..."
Per evitare questi dilemmi è sufficiente scrivere:

```
INPUT "ETA "; A
```

così sai e saprai sempre che il dato da introdurre nella variabile A è l'età.
Vediamo alcuni esempi di utilizzo dell'istruzione INPUT.

Esempi

```
INPUT A
```

Il tuo VIC 20 attende da te un numero che verrà assegnato automaticamente alla variabile A. Non ti informa però di che numero.

```
INPUT A$
```

Ora, invece, il tuo VIC 20 attende una stringa che verrà assegnata automaticamente alla variabile A$.

```
INPUT "NOME DI CITTA "; V$
```

La risposta che devi dare al computer è una stringa costituita dal nome di una città, ad esempio MILANO.
È sottinteso che un errore di digitazione (MILNO) non viene individuato dal tuo VIC 20: per lui è un testo più

che accettabile. Lo stesso vale per i numeri. Se gli fornisci un valore errato, o meglio diverso da quello che avresti dovuto o voluto introdurre, il tuo VIC 20 farà le sue elaborazioni su questo, fornendoti alla fine un risultato sbagliato. Attento quindi all'introduzione corretta dei dati.

```
INPUT "DAMMI 3  NUMERI"; A, B, C
```

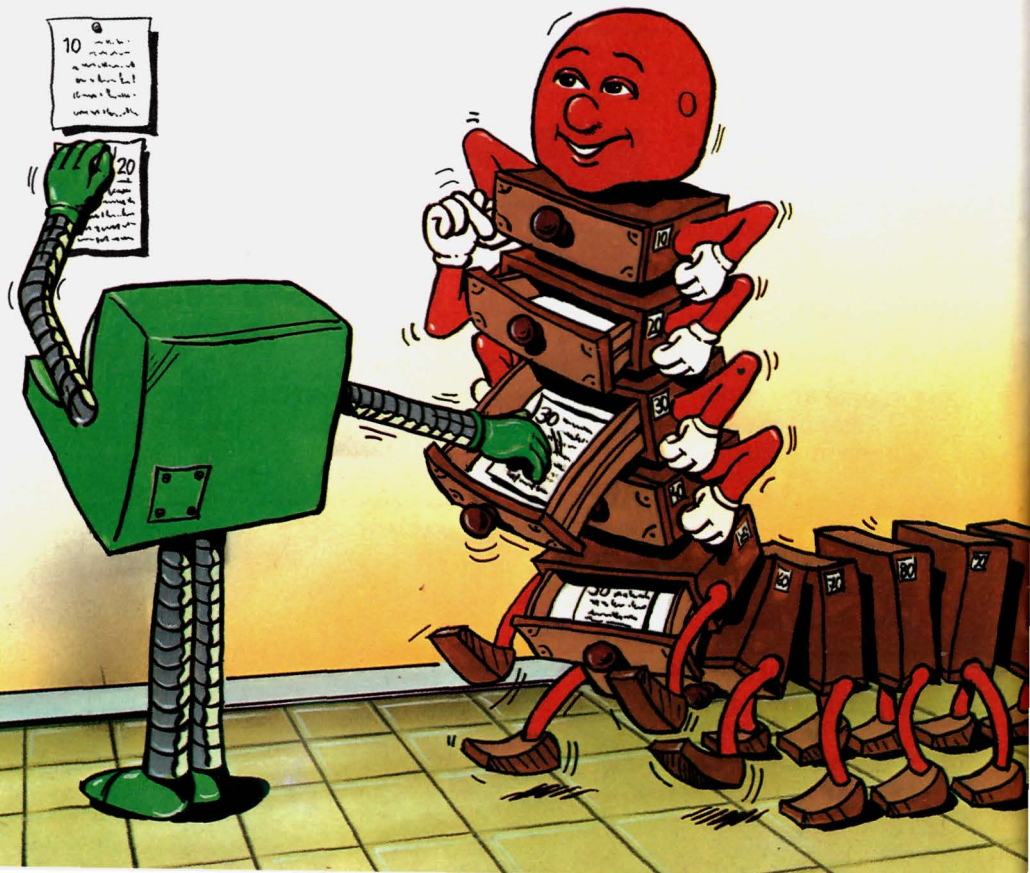
Alla comparsa del punto di domanda devi introdurre il primo valore numerico seguito da RETURN.
Appariranno?? (due punti di domanda). Introduci ora il secondo valore seguito sempre da RETURN; così per il terzo.
Puoi procedere anche in altro modo?
Te l'ho appena detto. Se non ti ricordi riguarda l'inizio di INPUT.

Sintassi dell'istruzione

INPUT ["Messaggio";] variabile [,variabile ...]

LIST

Visualizza sullo schermo video il programma presente in quel momento nella memoria centrale del tuo VIC 20. Il testo del programma, cioè l'insieme di tutte le sue linee scritte in linguaggio BASIC, prende il nome di listato. L'istruzione LIST permette appunto di listare un programma. I programmi BASIC hanno sempre le loro linee numerate per cui è possibile visualizzare tutto il programma o una parte di esso indicando la prima linea da listare e l'ultima (linea$_1$ - linea$_2$). LIST senza alcuna indicazione di linea lista tutto il programma.

Indicando un solo numero di linea (LIST linea) si visualizza solo quella linea. Per listare una parte del programma basta indicare le linee estreme separate da un trattino (LIST linea$_1$ - linea$_2$). Se si omette il numero linea$_1$ si lista tutto il programma dal suo inizio sino alla linea$_2$ (LIST - linea$_2$). Analogamente se si omette il numero linea$_2$ si lista da linea$_1$ sino alla

fine (LIST $linea_1$-).
LIST si usa normalmente in modo immediato quando si vuole controllare quale programma è presente in memoria centrale, oppure quando si sta scrivendo, e modificando, un nuovo programma. Per questo motivo sarebbe più corretto chiamarla "comando".
LIST potrebbe essere usata in modo differito, all'interno cioè di un programma, facendo listare automaticamente il proprio programma al momento della sua esecuzione.
Se il listato che vuoi visualizzare è più lungo di quanto lo schermo può mostrarti, il tuo VIC 20 fa scorrere automaticamente le linee dal basso verso l'alto arrestandosi all'ultima. Questo movimento è detto scroll.
Per ridurre la velocità è sufficiente tenere premuto il tasto CTRL.
Per interrompere, invece, completamente lo scroll devi ricorrere al tasto RUN/STOP. Da lì non potrai più continuare a scorrere il listato se non dando un nuovo comando LIST.

Esempi

LIST 10	Il tuo VIC 20 stampa a video solo la linea numerata con 10. Nel caso in cui il programma non contenga la linea numero 10, non stampa niente, rispondendo <ready>.
LIST	Questo comando lista l'intero programma in memoria. Ricordati che per rallentare lo scrolling devi tenere premuto il tasto CTRL.
LIST 150 -	Il listato viene visualizzato a partire dalla linea 150 sino alla fine.
LIST - 150	Visualizza dalla prima linea a quella numerata con 150, compresa.
LIST 120-150	Il tuo VIC 20 fa apparire a video le linee del programma tra la 120 e 150 comprese.

Sintassi dell'istruzione/comando

LIST { linea
- linea
linea -
$linea_1$ - $linea_2$ }

RUN

Istruzione del BASIC che pone in esecuzione il programma presente nella memoria centrale.

È più opportuno però chiamarla comando in quanto viene data normalmente in modo diretto senza precederla da alcun numero di linea.
L'indicazione numero linea è opzionale. Se manca, come nei casi più frequenti, il programma viene eseguito dall'istruzione di numero più basso, qualunque essa sia. Se viene indicato invece un numero di linea, il programma inizia da quella. In tale caso deve avere però un senso logico fare iniziare un programma da un punto diverso dall'inizio.
L'istruzione RUN cancella (resetta) ogni possibile valore delle variabili già avuto in precedenza.
In altre parole se si tenta di fare proseguire con RUN linea, l'esecuzione di un programma che si è fermato per un qualunque motivo (linea è il numero di linea da cui proseguire), si perde il contenuto di tutte le variabili.

Esempi

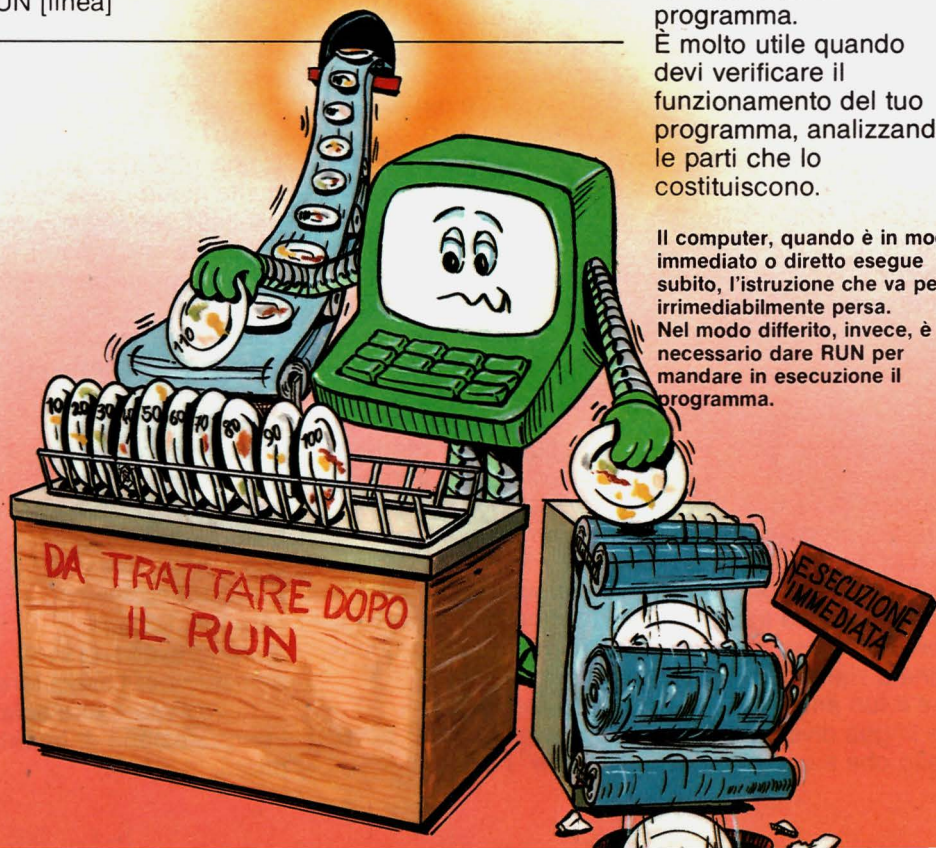
RUN	L'esecuzione del programma in memoria centrale parte dalla prima riga.
RUN 140	L'esecuzione del programma parte dalla riga 140 qualunque sia la sua posizione all'interno del listato.

Sintassi dell'istruzione/comando

RUN [linea]

END

Termina l'esecuzione del programma. Il calcolatore torna in modo "immediato" pronto ad accettare comandi da tastiera. L'istruzione END può essere posta in qualunque punto del programma (fine logica dell'esecuzione). In ogni caso può essere tranquillamente omessa ed è implicita dopo l'ultima linea del programma.
È molto utile quando devi verificare il funzionamento del tuo programma, analizzando le parti che lo costituiscono.

Il computer, quando è in modo immediato o diretto esegue subito, l'istruzione che va però irrimediabilmente persa. Nel modo differito, invece, è necessario dare RUN per mandare in esecuzione il programma.

Esempi semplici, anzi elementari

Supponi di saper compilare da solo *la* dichiarazione dei redditi, ma di non averne il tempo. Supponi anche di avere un amico che non ha la benché minima idea di aliquote, IRPEF, ILOR..., ma che dispone di tutto il tempo *necessario. Se vuoi* avere il suo aiuto nella preparazione del modello 740, dovrai indicargli su un foglio tutto quello che deve fare:

- guarda quanto ho guadagnato;
- scrivi *il risultato nel* foglio 5 riga 12;
- detrai il 10%;
- scrivi il risultato nel foglio 2 riga 9;
- cerca le fatture;
- scrivi l'importo sul foglio 3;
- ecc...

Il tuo amico però non ha le idee molto chiare. E difatti comincia a chiedersi dove può andare a prendere il modulo. Poi non capisce cosa vuoi dire con "detrarre", dove trovare le fatture e dove sia *segnato l'importo.*

Tu e il tuo amico parlate due lingue diverse e quello che a uno sembra facile (perché ha le conoscenze necessarie) per l'altro non è facile affatto.

Tuttavia, nonostante *l'insuccesso iniziale,* cercate un'altra soluzione. Ci sono due soli modi di risolvere il problema:

1° insegnare all'amico tutto quello che non sa;
2° cercare di guidarlo in modo tale che le *istruzioni gli siano* chiare.

Il primo metodo, anche se ragionevole, non è applicabile: se hai il tempo di insegnargli a compilare la dichiarazione dei redditi la faresti da solo.

Scegli così la seconda strada e, dopo un certo periodo di disagi, scopri che questo metodo non solo è più pratico, ma anche molto più veloce.

Con un calcolatore le cose sono molto simili: basta sostituire il "tuo *amico ignorante" con un* "computer" per renderti conto di quali siano i problemi che ogni programmatore deve saper affrontare.

Abbiamo visto che il problema di base è farsi capire: quali sono quindi, *in generale,* le operazioni che dobbiamo fare per poter "istruire" l'ignaro esecutore?

La prima cosa che devi accertare è qualè il problema da risolvere. Nel caso specifico, la dichiarazione dei redditi, abbiamo un problema ben definito, rigidamente determinato da regole che bisogna seguire. È bene precisare tuttavia che non sempre è così: molte volte bisogna trovare un modo in cui il problema è definito da uno schema generale, e trovarlo non è sempre facile.

Come risolverlo? Quali tecniche usare in sostanza per dare una risposta a quello che cerco?

E *una volta risolto, in* quale modo presentare i risultati? Nel nostro caso ne esiste uno solo: compilare correttamente il modulo.

Esistono però altri problemi che hanno più soluzioni.

La risposta a tutte queste domande ha un nome ben preciso: algoritmo.

L'algoritmo non è altro che quell'insieme di istruzioni con le seguenti caratteristiche:

- non devono essere *ambigue per l'esecutore:* ci deve essere una sola interpretazione di quanto ordini;

- devono portare ad un risultato definito: il prodotto dell'algoritmo;
- devono essere autosufficienti: bastano da sole a portare a termine l'algoritmo;
- devono essere implicitamente definite: ogni comando deve essere elementare (non ulteriormente scindibile in parti) e il significato di ogni operazione è vero per definizione.

Quanto detto ha valore generale: vale per il tuo amico quanto per il calcolatore. Se vuoi far risolvere un problema dal tuo VIC 20 devi fornirgli delle istruzioni con le caratteristiche enunciate.
Grazie a queste il tuo computer è in grado di risolvere un compito qualunque sotto forma di istruzioni elementari (è il linguaggio del computer detto appunto linguaggio macchina), ben definite e facilmente eseguibili.
Tuttavia, se dovessi fare sempre così (trovare cioè tutte le istruzioni elementari) per poter scrivere un algoritmo impiegheresti troppo tempo. È proprio per questo che sono stati realizzati i linguaggi di programmazione: per permettere a chi programma di usare una "lingua" sintetica che il computer traduce automaticamente nelle istruzioni elementari (linguaggio macchina) in grado di eseguire.
Come ogni lingua, ogni linguaggio di programmazione (compreso il BASIC di cui ci occupiamo in questo corso) ha la sua grammatica e la sua sintassi.
Innanzi tutto esso

Le fasi di un programma rispecchiano la costituzione fisica del tuo computer. C'è una fase di INPUT che coinvolge i dispositivi di INPUT, una di elaborazione svolta dall'unità centrale, una di OUTPUT demandata alle periferiche specifiche per la visualizzazione dei dati elaborati.

contiene affermazioni imperative in modo che non risultino ambigue e siano associate ad una azione ben precisa.
Contiene poi delle regole che prescrivono in che modo e in quale ordine deve scriverle.

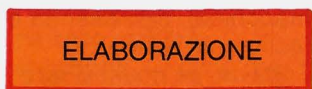
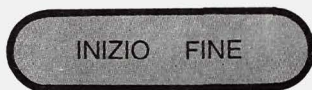
I diagrammi a blocchi

Soprattutto è difficile, una volta trovato l'algoritmo (la descrizione completa e accurata dei passi necessari alla risoluzione del problema), riuscire a verificarne rapidamente la validità.
Per questo ci si avvale di una rappresentazione grafica dell'algoritmo detta diagramma di flusso o diagramma a blocchi, in inglese flow chart. La simbologia usata è la seguente:
inizio/fine dell'algoritmo;
operazione di elaborazione dati;
operazione di scelta in base a una situazione;
operazione di ingresso/uscita.

INIZIO FINE

ELABORAZIONE

DECISIONE

INGRESSO USCITA

Anche con un linguaggio che ci solleva dal lavoro di traduzione al linguaggio della macchina (chiamato appunto linguaggio macchina o assembler) il lavoro da svolgere è notevole. È particolarmente difficile per esempio trovare un procedimento generale e definito laddove non ci sia. A questo scopo è stato introdotto il cosiddetto diagramma di flusso (dall'inglese flow chart).
Esso, come si può intuire dal termine, ci permette di vedere attraverso uno schema come si svolge un algoritmo, cioè qual è il suo flusso.
Ognuna di queste figure, collegate tra loro da una linea con freccia che indica la direzione del flusso, contiene una istruzione dall'algoritmo.
In conclusione: un algoritmo può essere rappresentato da un diagramma a blocchi o flow chart.
Vi sono altri simboli oltre ai quattro mostrati, questi però sono per ora più che sufficienti.
Tuttavia quando si parla della soluzione di un problema in questo modo sentite dire che si sta programmando quel problema e non "algoritmando". Perché? La differenza è sottile ma importante.
Abbiamo detto che un algoritmo serve a risolvere un problema, cioè esso ci dà i mezzi per dare una risposta.
Ma ci serve solo questo per poter far risolvere un problema a una macchina? La risposta è no. L'algoritmo è basilare perché indica alla macchina come risolvere il problema, ma noi dobbiamo "organizzare" un algoritmo in modo che dia la risposta corretta.

Scrivere il programma

Supponiamo di aver trovato il procedimento per trasformare le miglia in chilometri e che tuttavia ci serve un calcolo espresso in miglia: come facciamo? Escludiamo l'algoritmo di conversione quando ci serve che non operi. Ma questo chi lo dice? Il programma.

Il programma è quindi una serie ordinata di algoritmi organizzata in maniera da definirne la sequenza (l'ordine di esecuzione). Ora finalmente siamo in grado di programmare. Vediamo di fare un semplice programmino che calcoli il perimetro di un quadrato qualsiasi. All'inizio ti chiederà la misura del lato, poi ti mostrerà quella del perimetro.

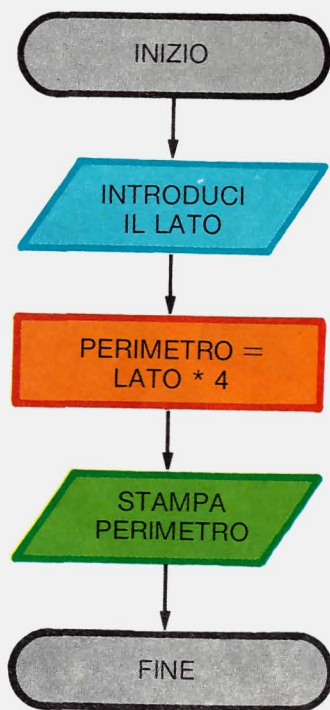
Sostanzialmente ti chiede una informazione in entrata (introduci il lato) e ti fornisce un risultato (perimetro).

In generale, visto che un programma è una successione di algoritmi, esso dovrà avere almeno un dato in entrata e uno in uscita. Genericamente questo processo, definito di input/output (ingresso/uscita), è la prima fase della soluzione di un problema tramite calcolatore (vale a dire

INIZIO

INTRODUCI IL LATO

PERIMETRO = LATO * 4

STAMPA PERIMETRO

FINE

attraverso quali variabili il problema si può risolvere).
La fase immediatamente successiva è il "come" utilizzare le variabili trovate nell'I/O, cioè scrivere l'algoritmo.
La terza ed ultima fase è trovare una forma conveniente per l'uscita dei dati (output). Se risolvi questi punti potrai programmare qualsiasi problema.
È consigliabile (anche se non necessario) analizzare il problema nell'ordine esposto in modo che tutti i passi da fare siano legati tra loro facilitando la messa a punto del programma.
Nel programmino di prima perciò devi stabilire:
1) Quante e quali variabiliti servono? Devi fare in modo che il programma richieda i dati necessari (in questo caso uno solo: il lato).
2) Che procedimento usi?
Un procedimento che ti permetta di calcolare il perimetro (lato x 4).
3) Come presenti il risultato (esplicito)?
Stampandolo sullo schermo.
Come puoi intuire il problema non è imparare su quali istruzioni si basa il funzionamento di un computer, ma ridurre il tuo obiettivo in forma tale da essere "digeribile" dal computer.
Programmare non significa quindi solo la perfetta conoscenza di quello che la macchina

I FASE
INPUT

II F
ELABO

può offrire, ma implica anche una profonda comprensione del problema da affrontare. Non solo perciò è necessaria una conoscenza "informatica" cioè strettamente legata ai calcolatori, ma anche una conoscenza dei metodi che è utile usare per ridurre il problema nei termini accettabili dal computer (computabile). Per confezionare buoni programmi occorre perciò un procedimento analitico, in grado cioè di risolvere un problema in forma logica.

Questa è la qualità indispensabile ad ogni buon programmatore. È meglio precisare che il flow chart come metodo per risolvere un problema è stato soppiantato dalla programmazione strutturata. Più il problema è complesso, infatti, più si complica il diagramma di flusso che ne rappresenta l'algoritmo. Questo ovviamente comporta vari inconvenienti, ma soprattutto porta ad avere un programma poco leggibile e difficilmente modificabile. Il flow chart rimane, comunque, un ottimo metodo per illustrare quello che il programma è in grado di fare, anche se da solo è insufficiente alla realizzazione del programma.

Ricapitolando:

- usa il flow chart per illustrare un programma e non per scriverlo;
- pensa per modelli, cioè crea una procedura generale logica valida per risolvere il problema;
- utilizza i diagrammi di blocchi per correggere i programmi troppo contorti: dal disegno potrai facilmente vedere se il programma, pur elaborando i dati in maniera corretta, risulta più complesso del necessario (vedremo più tardi in dettaglio questa tecnica).

ASE
RAZIONI

III FASE
OUTPUT

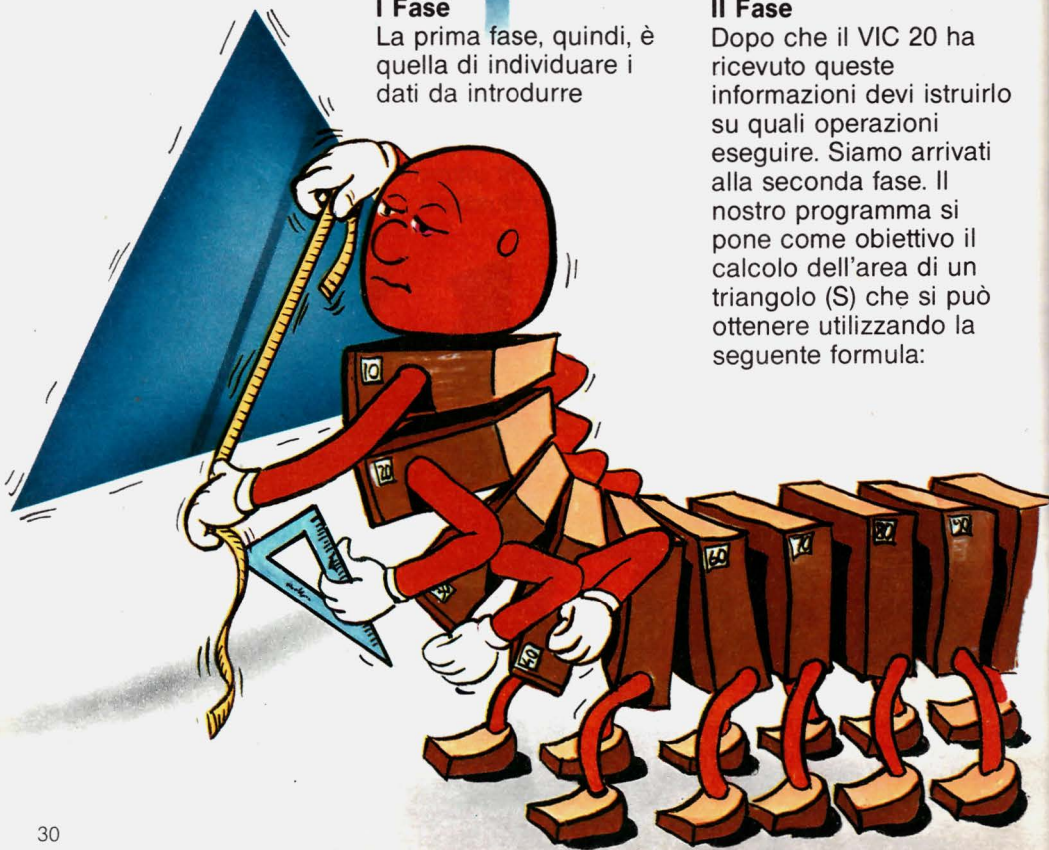
Il tuo primo programma

Sei appena agli inizi ed è poco che studi con me il BASIC e la programmazione, ma hai già tutti gli elementi per scrivere un semplice ma funzionante programma. Ti guiderò passo passo anche se presto ti renderai conto che ciò non sarà più necessario. Scriviamo insieme un programma che calcoli l'area di un qualsiasi triangolo.
Come si è detto più volte, il tuo VIC 20 può soltanto elaborare i dati che tu gli fornisci.

I FASE
INPUT

II FASE
ELABORAZIONI

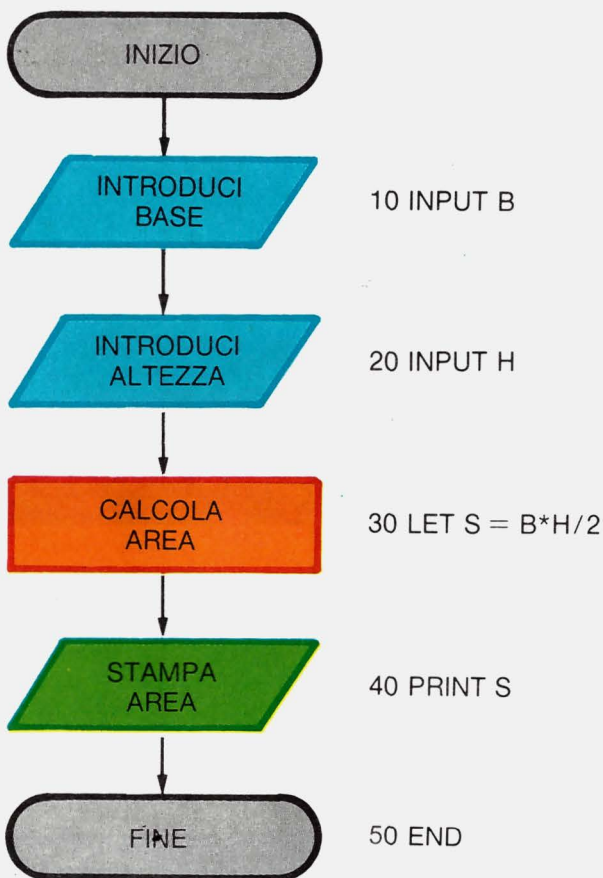
I Fase
La prima fase, quindi, è quella di individuare i dati da introdurre necessari al calcolo dell'area di un triangolo. Come ben sai occorre conoscere la base (B) e l'altezza (H) che per il tuo computer sono variabili numeriche reali.

II Fase
Dopo che il VIC 20 ha ricevuto queste informazioni devi istruirlo su quali operazioni eseguire. Siamo arrivati alla seconda fase. Il nostro programma si pone come obiettivo il calcolo dell'area di un triangolo (S) che si può ottenere utilizzando la seguente formula:

$$S = \frac{\text{Base x Altezza}}{2} = \frac{\text{B x H}}{2}$$

Devi quindi fornire al tuo computer questa formula affinché possa procedere all'elaborazione.

III FASE
OUTPUT

III Fase
Il tuo VIC 20 ha ora tutti gli elementi per procedere al calcolo dell'area di un triangolo. Il risultato, però, se lo tiene per sé se non gli dici come rendertelo disponibile. Siamo arrivati all'ultima fase, quella di OUTPUT, in cui il computer deve ritornarti il risultato delle elaborazioni, cioè l'obiettivo del programma.
Traduciamo ora quanto detto a parole in un diagramma a blocchi con a fianco le istruzioni BASIC che conosci.

Diagramma a blocchi	Istruzione BASIC
INIZIO	
INTRODUCI BASE	10 INPUT B
INTRODUCI ALTEZZA	20 INPUT H
CALCOLA AREA	30 LET S = B*H/2
STAMPA AREA	40 PRINT S
FINE	50 END

Non è stato molto difficile. Vero?

VIDEOESERCIZI

Annota nello spazio apposito il risultato da te prévisto per ciascun esercizio proposto e poi verifica la soluzione col tuo VIC 20. Se avrai commesso anche un solo errore ripassa la lezione.

```
PRINT "Introduci un numero"; : INPUT N
```

```
INPUT "Introduci un numero"; N
```

```
10 PRINT "BIT";
20 PRINT "PERSONAL SOFTWARE"
30 PRINT "VIDEOGIOCHI"
40 END
```

```
10 PRINT "BIT";
20 END
30 PRINT "PERSONAL SOFTWARE"
40 PRINT "VIDEOGIOCHI"
```

Per il programma precedente LIST 20

segue per lo stesso programma LIST

ancora per lo stesso programma RUN 20

e ora RUN